

# **HARDWARE DESIGN OF ELECTRONIC NOTICE BOARD**

- 1.DESIGN A DIGITAL THERMOMETER**
- 2.DESIGN A SYSTEM TO MEASURE  
TEMPERATURE AND HUMIDITY  
AND TO LOG THE DATA**

NAME: ASIT BARAN CHANDA  
Roll: 110EC0160



Department of Electronics & Communication engineering  
National Institute of Technology Rourkela  
Rourkela-769008, Odisha, INDIA

# **HARDWARE DESIGN OF ELECTRONIC NOTICE BOARD**

*A Thesis Submitted in partial Fulfillment  
Of the Requirements of the Degree of*

**“Bachelor of Technology”**

in

Electronics and Communication Engineering

by

ASIT BARAN CHANDA

Roll: 110EC0160

Under The Guidance

of

Prof. SARAT KUMAR PATRA



Department of  
Electronics & Communication  
engineering National Institute of  
Technology Rourkela  
Rourkela-769008, Odisha, INDIA

# National Institute of Technology, Rourkela



## DECLARATION

I hereby declare that the project work entitled “HARDWARE DESIGN OF ELECTRONIC NOTICE BOARD” is a record of my original work done under Prof. SARAT KUMAR PATRA, National Institute of Technology, Rourkela. I have followed the guidelines provided by the Institute while preparing the thesis. I have confirmed to the norms and guidelines in the Ethical Code of Conduct of the Institute. Throughout this project wherever contributions of others are involved, every endeavour has been made to acknowledge this clearly with due reference to literature. This project work is being submitted in the partial fulfilment of the requirements for the degree of Bachelor of Technology in Electronics and Communication Engineering at National Institute of Technology, Rourkela for the academic session 2010– 2014.



**Department of Electronics and Communication Engineering**  
**National Institute of Technology, Rourkela**

**C E R T I F I C A T E**

This is to certify that the Thesis report entitled “**HARDWARE DESIGN OF ELECTRONIC NOTICE BOARD**”, submitted to the National Institute of Technology, Rourkela by Mr. Asit Baran Chanda , Roll No: 110EC0160 for the award of Bachelor of Technology in Electrical Engineering is a bona-fide record of research work carried out by him under my supervision and guidance.

The candidate has fulfilled all the prescribed requirements. The report which is based on candidate's own work has not been submitted elsewhere for a degree/diploma. In my opinion, the report is of standard required for the award of a Bachelor of Technology in Electronics & Communication Engineering.

Prof. Sarat Kumar Patra

Supervisor

Department of Electronic & Communication Engineering

National Institute of Technology Rourkela – 769 008 (ODISHA)

## ACKNOWLEDGEMENTS

On the submission of my Thesis report of “*HARDWARE DESIGN OF ELECTRONIC NOTICE BOARD*”, I would like to express my sincere gratitude to my supervisor **Prof.Sarat Kumar Patra** for his guidance. I would like to thank him for his encouragement, and support throughout the course of this work from the core of my heart . It was an invaluable learning experience for me to be one of his students. As my supervisor his insight, observations and suggestions helped me to establish the overall direction of the research and contributed immensely for the success of this work. His immense knowledge, technical skills and human values have been a source of inspiration to me.

My thanks are extended to **Mr Pallab Majhi** who helped me with the roadblocks in my project.I would also like to thank him for providing me with all the necessary equipments required to complete this project.

Finally I would like to thank my friends, who built an academic and friendly research environment that made my study at NIT, Rourkela most fruitful.

**ASIT BARAN CHANDA**

**110EC0160**

## ABSTRACT

In this project a hardware is designed to measure the temperature and humidity and log the data in a SD card. At the core of the circuit is the ARDUINO UNO microcontroller which controls all its functions. A humidity and temperature sensor DHT11 is used for sensing the humidity and temperature of the environment and the system displays the humidity and temperature on an LCD in the range of 20%RH to 90%RH and 0°C to +50°C respectively. This humidity and temperature values are then programmed to log and saved in an external SD card as text file. The logged data can be seen by opening the text file in a computer.

# CONTENTS

Declaration.....	3
Certificate.....	4
Acknowledgement.....	5
Abstract.....	6
Contents.....	7
List Of Figures.....	9

## **Chapter 1 LITERATURE REVIEW .....10**

1.1 Introduction .....	11
------------------------	----

1.2 Hardware Requirements.....	11
--------------------------------	----

1.3 LM35.....	11
---------------	----

1.4 DHT11.....	13
----------------	----

1.5 ARDUINO UNO.....	16
----------------------	----

1.6 ARDUINO SD CARD SHIELD.....	17
---------------------------------	----

1.7 LCD DISPLAY (2 X 16).....	18
-------------------------------	----

## **Chapter 2 SYSTEM DESIGN AND IMPLEMENTATION.....22**

2.1 Hardware Block Diagram.....	23
---------------------------------	----

2.2 Implementation.....	24
-------------------------	----

2.3 Working.....	24
<b>Chapter 3 SOFTWARE AND CODE.....</b>	<b>26</b>
3.1 Introduction.....	27
3.2 Source Code.....	28
<b>Chapter 4 RESULTS AND CONCLUSION.....</b>	<b>38</b>
4.1 Conclusions.....	39
4.2 Future Scope and enhancements.....	40
<b>Chapter 5 REFERENCES.....</b>	<b>41</b>
5.1 References.....	42



## **LIST OF FIGURES**

1) Figure 2.1 LM35.....	12
2) Figure 2.2 Bottom View Of LM35.....	12
3) Figure 2.3 LM35 Connection.....	12
4) Figure 2.4 DHT11.....	14
5) Figure 2.5 DHT11 Pin Configuration and Connection.....	14
6) Figure 2.6 DHT11 data transmission.....	15
7) Figure 2.7 ARDUINO UNO Board.....	16
8) Figure 2.8 Arduino SD card shield.....	18
9) Figure 2.9 LCD Display.....	19
10) Figure 2.10 LCDPinDiagram.....	20
11)Figure 3.1 Hardware Block Diagram .....	22
12) Figure 3.2 Final setup of the project.....	24
13) Snapshot of the Arduino software.....	26

# **CHAPTER 1**

## **LITERATURE AND REVIEW**

## **1.1 Introduction**

A system is designed to measure the humidity and temperature in a room or in a building .So the sensor which is to be used should has a temperature range of 10°C to 40°C ,and humidity range of 25%RH to 80%RH. DHT11 is a sensor which qualifies in both ranges. The data is to be shown in a LCD display, so an alphanumeric LCD display will suffice . A microcontroller is required take sensor values from sensor and give command for displaying .For this an Arduino uno can be used . The data from this sensor is also logged in a memory. For this an Arduino SD card shield is required. Here DHT11 measures the humidity and temperature and sends an analog voltage to the the microcontroller ARDUINO UNO . Then data from 4 sensors are taken and the average is calculated by using a program written in arduino software. The averaged values of both the humidity and temperature is displayed in a LCD.

## **1.2 Hardware Requirements**

- 1) LM35
- 2) DHT11
- 3) ARDUINO
- 4) ARDUINO SD CARD SHIELD
- 5) LCD DISPLAY
- 6) DC ADAPTER

## **1.3 LM35**

The Lm35 arrangement is accuracy incorporated-circuit temperature sensors, with a yield voltage directly corresponding to the Centigrade temperature. So programmer does not requires to subtract

any factor from the output to obtain Centigrade scaling as in case of the other sensors which are calibrated in ° Kelvin . Amplification of output voltage is not required because LM35 generates high output voltage.

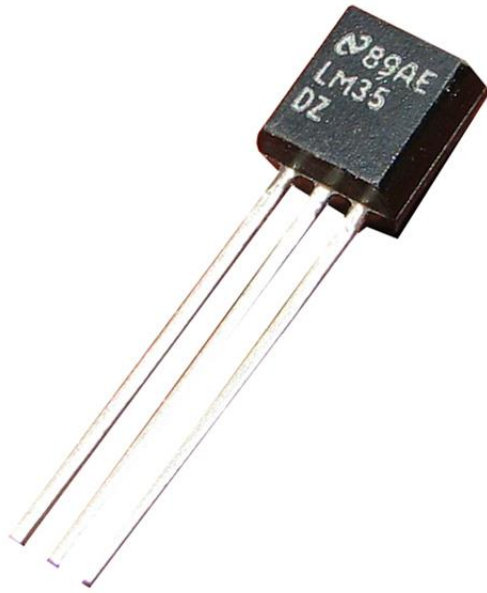


Figure 1.1 LM35

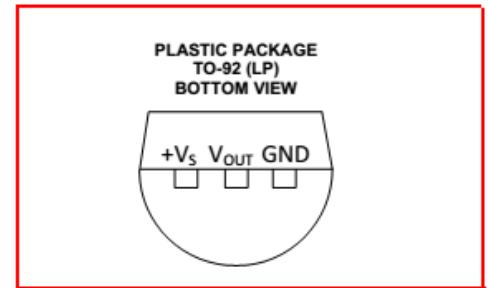


Figure 1.2 Bottom View Of LM35

External calibration or trimming is not required in case of LM35 .LM35 can provide typical accuracies upto  $\pm 0.25^{\circ}\text{C}$  over a range of  $[20^{\circ}\text{C} , 25^{\circ}\text{C}]$  and upto  $\pm 0.75^{\circ}\text{C}$  over a range of  $[-55^{\circ}\text{C} , +150^{\circ}\text{C}]$  .The gadget is used with single power supplies.

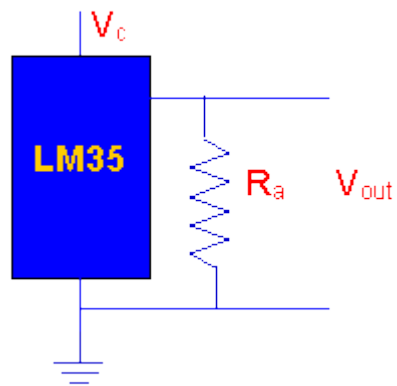


Figure 1.3 LM35 Connection

$V_c$  is supply voltage, it can vary from 4 to 30 v. 5v or 12 v are mainly v used values. In the above figure  $R_a$  is  $V_c / 10^{-6}$ , it can range from 80 KW to 600 KW, but mainly 80 KW is used. LM35 draws only a little current of 60  $\mu$ A from the power source, as a result its self-heating is limited to 0.1°C in still air. The sensor circuitry is constructed so that it wouldn't be inclined to oxidation.

LM35 has a sensitivity of 10mV / °C. It uses a conversion factor i.e. proportional, the conversion factor is 100 °C/V. The formula used to conversion of Volts to °C is

$$\text{Temperature (}^{\circ}\text{C)} = V_{\text{out}} * (100^{\circ}\text{C} / \text{V})$$

So if  $V_{\text{out}}$  is 0.5V, then, Temperature = 50 °C. Output voltage varies linearly with temperature.

## **1.4 DHT11**

This DHT11 humidity & temperature Sensor is calibrated with a digital signal output. It can ensure high reliability and excellent long-term stability by using the exclusive digital-signal-acquisition technique and humidity & temperature sensing technology. DHT11 incorporates a humidity measurement component which is resistive type and an negative temperature coefficient temperature measurement component. When connected to a excellent quality 8-bit microcontroller, shows quick response, non-interference ability. The whole setup is economical.

DHT11 sensors are very carefully calibrated in lab. These sensors are very precise on humidity calibration. The calibration coefficients are saved in the onetime programmable memory as programmes. These coefficients are used by the DHT11's internal signal while calculating humidity and temperature. The single-wire serial interface makes framework joining fast and simple. Dht11 works on a very low power, its size is very small and can transmit signal up-to-20 meter, because of these features DHT11 can be used for various applications.

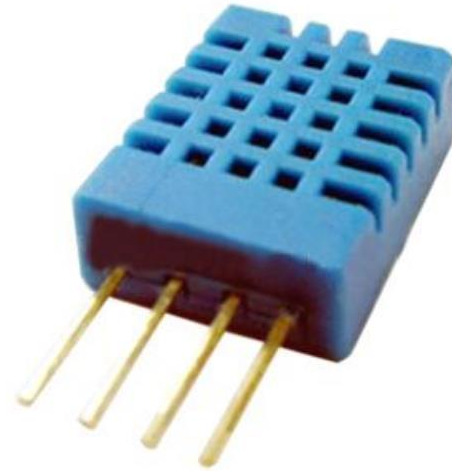


Figure 1.4 DHT11

DHT11 is shown in the above figure. Its specifications are

Humidity range – 20% RH to 90%RH with an accuracy of  $\pm 5\%$  RH

Temperature range – 0-50 °C with an accuracy of  $\pm 2^\circ\text{C}$

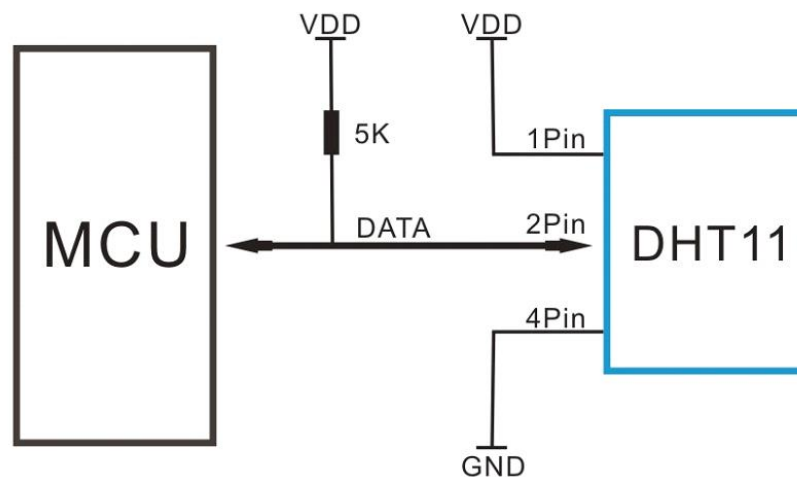


Figure 1.5 DHT11 Pin Configuration and Connection

Pin 1 is power supply pin. Power supply can vary from 3-5.5V DC. When the sensor is powered , no instructions should be given to the sensor upto one second so as to pass the unstable status. A capacitor of 50nF can be connected for power filtering purposes (between VDD and GND).

Pin 2 is data pin . One communication process lasts till 4ms. A complete data transmission is of 5 byte. The data has decimal and integral parts. Data format is , 1byte integral humidity data +

1 byte integral temperature data + 1 byte decimal humidity data + 1 byte decimal temperature data + 1 byte check sum. The transmitted data can be checked by adding the first four byte i.e be sum of "1 byte integral humidity data ,1byte integral temperature data, 1 byte decimal humidity data ,1 byte decimal temperature data", this should be equal to the last byte i.e check sum byte, if not equal then the transmitted data is wrong.

At first the microcontroller unit gives a start signal, Then the mode of DHT11 changes to the running-mode from the low-power-consumption mode. And then DHT11 waits for microcontroller unit for the completion of the start signal . DHT11 gives a response signal of 5 byte data after its completion. This includes the temperature and relative humidity information to MCU.

When the communication between MCU and DHT11 starts, the data bus voltage level is set from high to low by programme in the microcontroler and the data bus is kept low for at least 18ms to ensure that DHT11 has detected MCU's signal, then MCU pulls up voltage and waits 20-40us for DHT's response. Once DHT identifies the start signal, it transmits a low-voltage-level response signal, which lasts upto 80us, followed by a high voltage-level response signal which also lasts upto 80us. This is for DHT11's preparation for transmitting data.

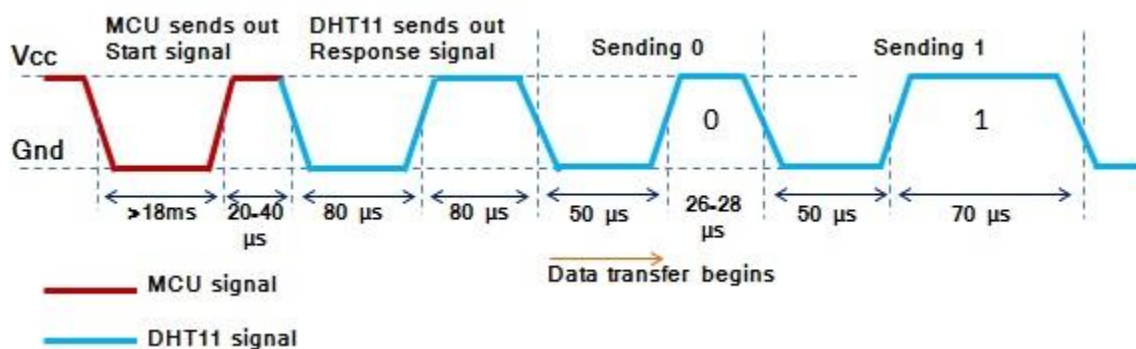


Figure 1.6 DHT11 data transmission

When the sensor transmits information to microcontroller, each bit of the data starts with the 50us low-voltage-level followed by a high voltage level. If the duration is 70us then the data bit is '1' or if the duration is 26-28us then the data bit is '0'.

## 1.5 ARDUINO UNO

Arduino Uno is a microcontroller board which consists of 6 analog input pins, 14 digital input/output pins, a power jack, an ICSP header, a power jack, a USB connection, a 16 MHz crystal oscillator, and a reset button. It is powered by Atmega328 microcontroller. It contains everything needed to support the microcontroller. It is only required to be connected to a laptop via a USB cable or powered with a AC-to-DC adapter or battery to get started. The Uno differs from all previous sheets in that it doesn't utilize the FTDI USB-to-serial driver chip. Rather, it emphasizes the Atmega8u2 modified as a USB-to-serial converter.

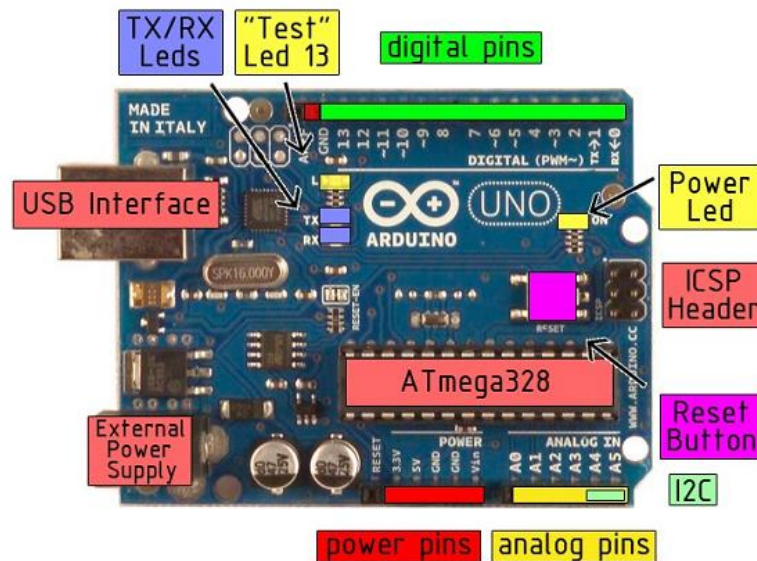


Figure 1.7 ARDUINO UNO Board

Power supply to Arduino can be given through USB connection or from an external power supply. External power can come either from an AC-to-DC adapter (2.1mm center-positive plug) or from



a battery. Leads from a battery can be inserted in the Gnd and Vin pin headers of the power connector.

Arduino board operates on an external supply of 6 to 20 volts. If it is supplied with a voltage less than 7V then the 5V pin may supply less than 5V and the board may become unstable. If it is supplied with a voltage more than 12V, then the board gets damaged due overheating of voltage regulator . Thus 7 to 12 volts is recommended range.

The ATmega328 has 32 KB of Flash memory . It also has 2 KB of SRAM and 1 KB of EEPROM. The 14 digital pins on the board can be used as an input or output . These pins operate at 5 volts . The DC current per input/output pin is about 40 mA and the pins have an internal pull-up resistor of 20-50 kOhms.

The board has 14 digital input/output pins out of which 6 provide PWM output (pin 3, pin 5, pin 6, pin 9, pin 10 and pin 11). The DC current for 3.3V pin is about 50 mA.

The Arduino board has 6 analog inputs, each of which provides 10 bits of resolution or 1024 different values. By default they measure from ground to 5 volts.

Its clock speed is about 16MHz.

The Arduino Uno can be programmed with the Arduino software.

## **1.6 ARDUINO SD CARD SHIELD**

An Arduino SD card shield is an extension of the Arduino board which contains an extra feature i.e a SD card slot . This is used for storing data from the Arduino.

Microcontroller and the SD card communicate using SPI, which takes place on digital pins 11, 12, and 13 ..

Pin 11 - MOSI

Pin 12- MISO

Pin 13 – clock

Pin 10 – Chip Select

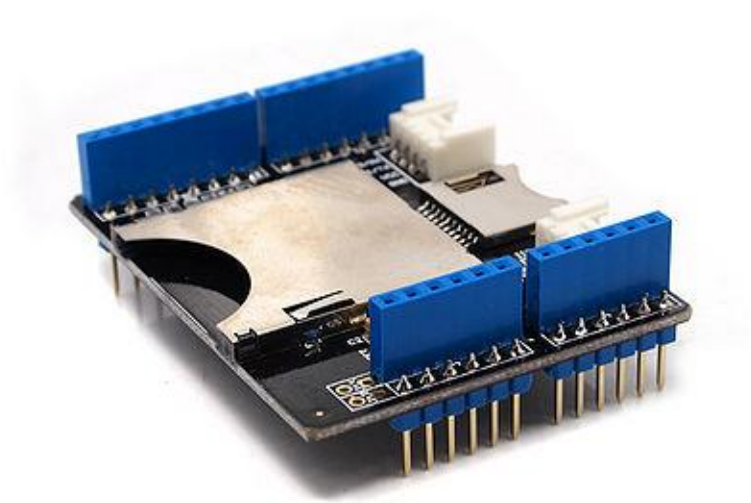


Figure 1.8 Arduino SD card shield

The shield is such that its pins go directly through the pin slots of Arduino uno.

## 1.7 LCD Display



Figure 1.9 LCD Display

LCD (Liquid Crystal Display) is an electronic display module and has a vast range of applications. A 16x2 LCD display is a very basic module and is commonly used in various devices and circuits. These modules are preferred over the seven segments and other multi segment LEDs.

In a LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command registers and Data registers .

The command register stores the command instructions given to the LCD. A command is an instruction given to the LCD which does predefined tasks like initializing it, controlling display, clearing its screen, setting the cursor position, etc. The data register stores the ASCII value of the character as data and this is displayed on the LCD.

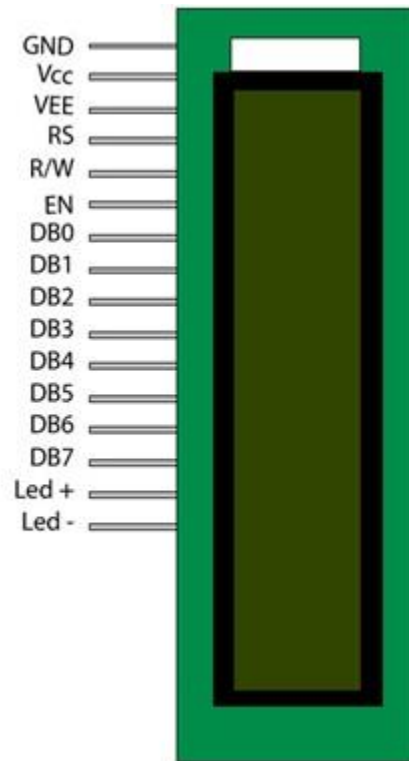


Figure 1.10 LCD Pin Diagram

### LCD pin configuration

Pin1 Ground - Ground (0V)

Pin2 Vcc - Supply voltage; 5V (4.7V – 5.3V)

Pin3 V<sub>EE</sub> - Supply voltage; 5V (4.7V – 5.3V)

Pin4 RS - Selects command register when low; and data register when high

Pin5 R/W - Low to write to the register; High to read from the register

Pin6 EN - Sends data to data pins when a high to low pulse is given

Pin7 to Pin14 - 8 bit data pins

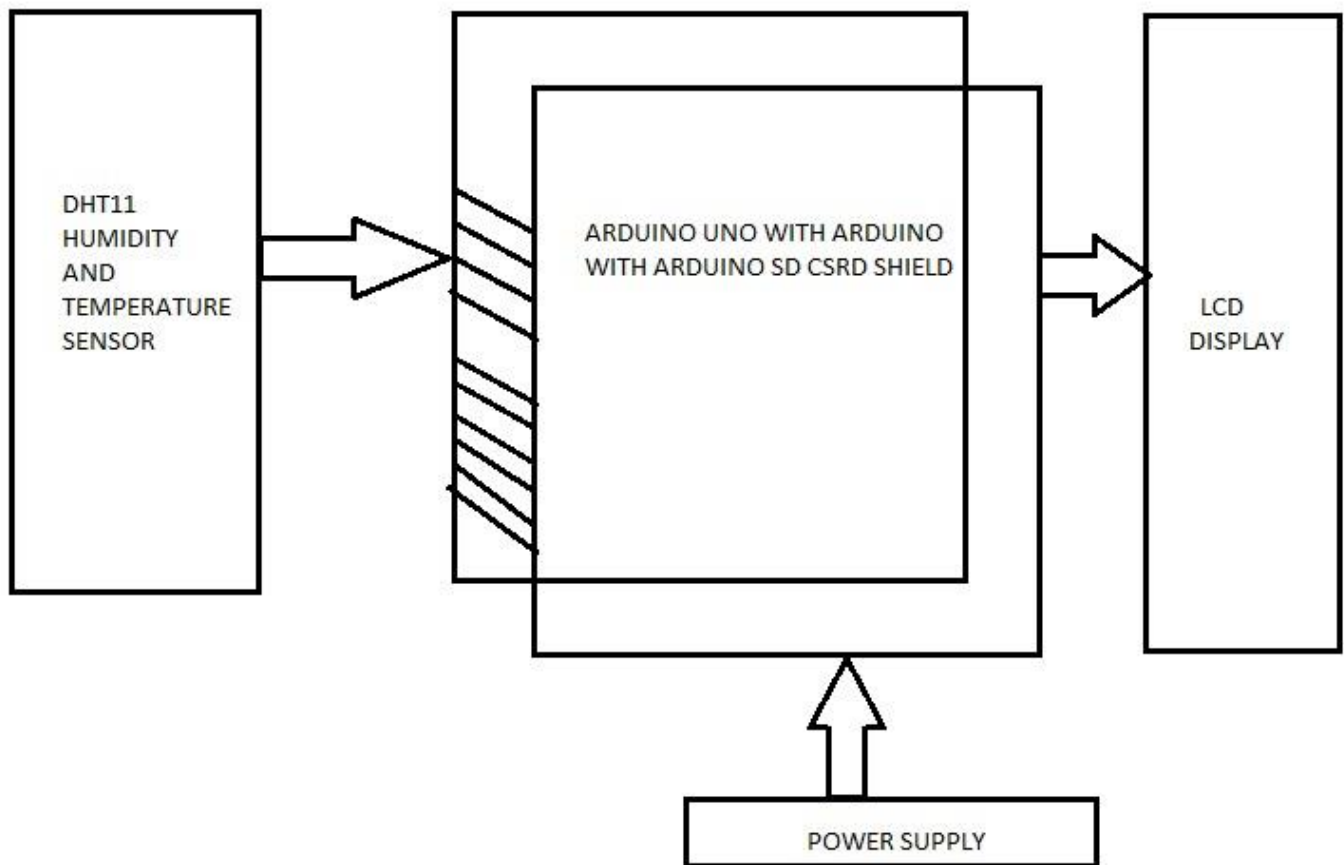
Pin15 led+ - Backlight V<sub>CC</sub> (5V)

Pin16 led- - Backlight Ground (0v)

# **CHAPTER 2**

## **SYSTEM DESIGN AND IMPLEMENTATION**

### 3.1 Hardware Block Diagram



2.1 Hardware Block Diagram

### **3.2 Implementation**

- 1) The data pins of the four sensors are connected to four analog inputs of Arduino board .  
Sensor 1 to port A5, sensor 2 to port A4, sensor 3 to port A3 and sensor 4 to port A2.
- 2) An Arduino SD card Shield is connected to the board.
- 3) LCD is connected to Arduino through digital pins 2 to 5, DB7 pin to digital pin 2 , DB6 pin to digital pin 3 , DB 5 pin to digital pin 4, DB4 pin to digital pin 5.
- 4) Arduino is USB powered , the program is run and the output is observed in the LCD and the in the SERIAL MONITOR of Arduino software.

### **3.3 Working**

The project uses a humidity and temperature sensor DHT11 which is capable of sensing the relative humidity and temperature in Celsius. It gives an output of 40 bits to the analog pins of Arduino Uno. These are converted into humidity and temperature and are collected via a programme . The output data is logged into a SD card using Arduino SD card shield . LCD is interfaced with the microcontroller & the value of the averaged humidity and temperature are displayed on the LCD.



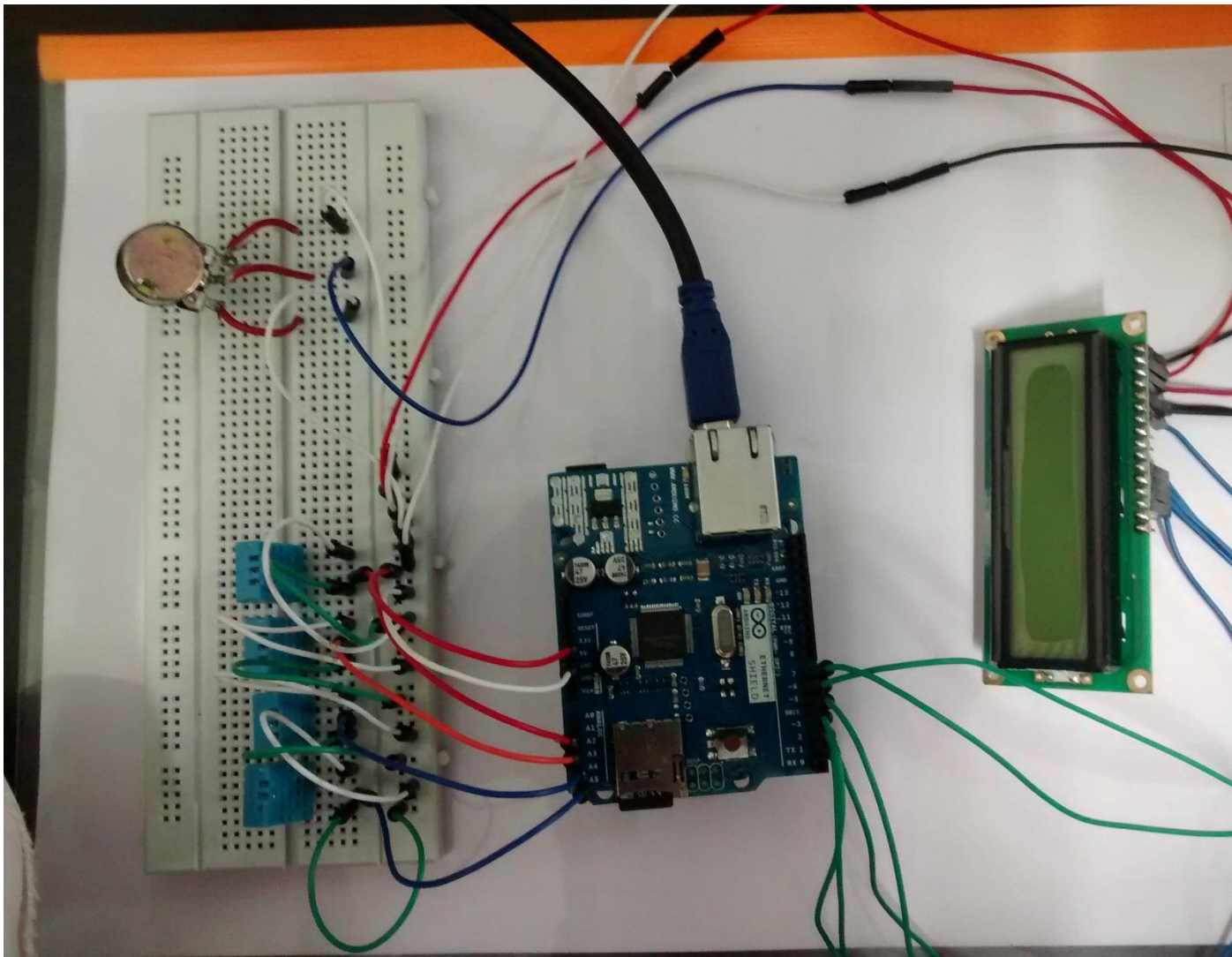


Figure 2.2 Final setup of the project

## **Chapter 3**

# **SOFTWARE AND CODE**

### 3.1 Introduction

The Arduino uno is programmed in an Arduino software . This is an open source software. The open-source Arduino environment makes it easy to write code and upload it to the Arduino board. It is compatible with Windows, Mac OS X, and Linux.

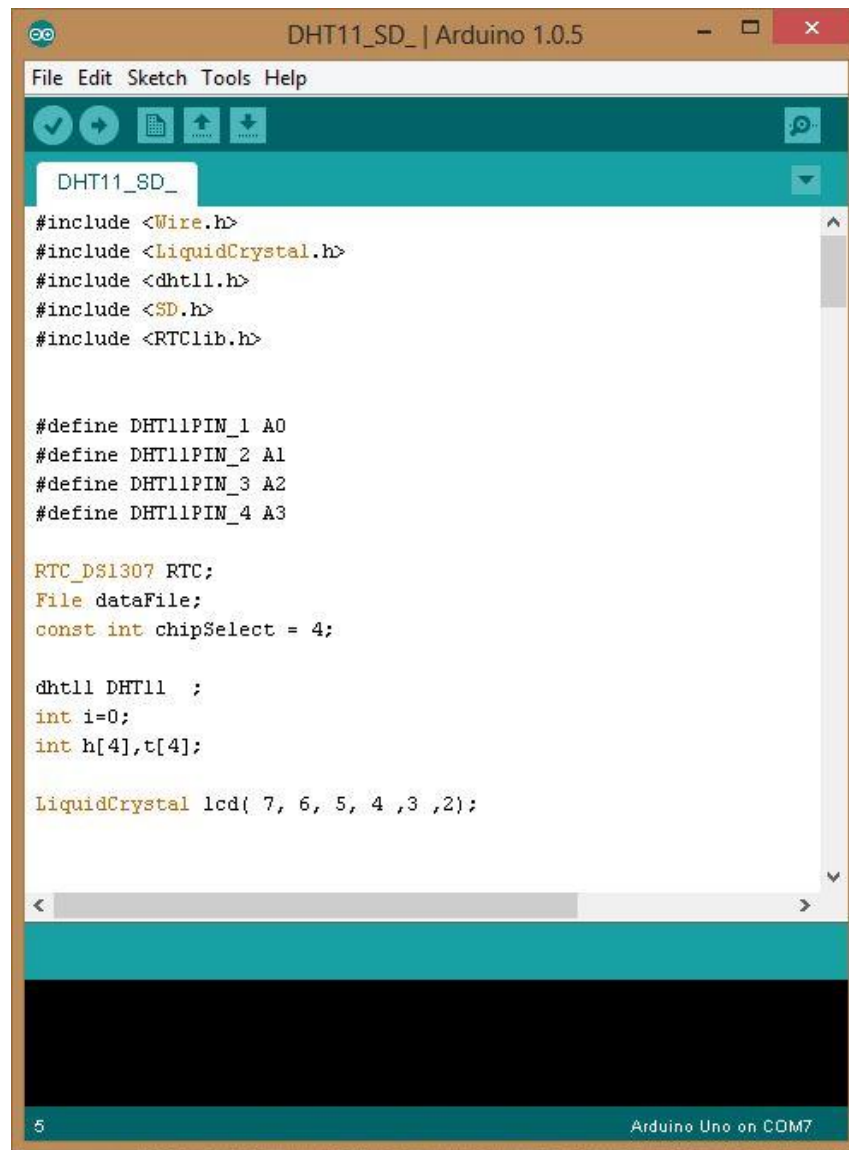


Figure 3.1 Snapshot Of The Arduino Software

## **3.2 SOURCE CODE**

### **1) PROGRAM TO MEASURE TEMPERATURE FROM LM35**

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(7 , 6, 5, 4 ,3 ,2);
```

```
int inputpin = 0;
```

```
int temp = 0,temperature=0;
```

```
int tsample[4];
```

```
int maxt = -100,mint = 100;
```

```
int i;
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
lcd.begin(16, 2);
```

```
lcd.clear();
```

```
lcd.setCursor(3,0);
```

```
lcd.print("FINAL YEAR" );
```

```
lcd.setCursor(5,1);
```

```
delay(1000);
```

```
lcd.print("PROJECT");
```

```
delay(1000);
```

```

lcd.clear();

}

void loop()

{

    for(i = 0;i<=3;i++)

    {

        temp = 5.0 * analogRead(inputpin) * 100.0 / 1024.0;

        tsample[i] = temp;

        temperature = temperature + tsample[i];

        delay(500);

    }

    temperature = temperature/4.0;

    if(temperature > maxt)

        {   maxt = temperature;   }

    if(temperature < mint)

        {   mint = temperature;   }

    Serial.print( " CURRENT T : " );

    Serial.print(temperature,DEC);

    Serial.println(" Celsius ");

    lcd.clear();

    lcd.setCursor(1,0);

```

```
lcd.print("CURRENT T: ");

lcd.print(temperature,DEC);

lcd.print("oC ");

Serial.print("MAXIMUM T: ");

Serial.print(maxt,DEC);

Serial.println(" Celsius ");

lcd.setCursor(0,1);

lcd.print("MAX-");

lcd.print(maxt,DEC);

lcd.print("oC ");

Serial.print("MINIMUM T: ");

Serial.print(mint,DEC);

Serial.println(" Celsius ");

Serial.print("\n");

lcd.setCursor(8,1);

lcd.print("MIN-");

lcd.print(mint,DEC);

lcd.print("oC");

tempc = 0;

delay(2000); //

}
```

*2)PROGRAM TO MEASURE TEMPERATURE AND HUMIDITY  
FROM A DHT11 SENSOR AND LOG THE VALUES*

```
#include <SD.h>

#include <dht11.h>

#include <Time.h>

#include <LiquidCrystal.h>

#define DHT11PIN_1 A5

#define DHT11PIN_2 A4

#define DHT11PIN_3 A3

#define DHT11PIN_4 A2

const int interval = 1*1000;

long lst = 0;

dht11 DHT11 ;

int i=0;

float h[4],t[4], maxh=0, maxt=0 , minh=100, mint=100;

const int chipselect = 4;

LiquidCrystal lcd( 7,6,5,4,3,2);

void setup() {

    Serial.begin(9600);

    setTime(10,0,0,9,5,14);

    lcd.begin(16, 2);

    lcd.clear();
```

```

lcd.setCursor(3,0);

lcd.print("FINAL YEAR" );

lcd.setCursor(5,1);

delay(1000);

lcd.print("PROJECT");

delay(1000);

if (!SD.begin(chipselect))
{
  Serial.println("Card failed, or it is not present");
}
else {
  Serial.println("card is initialized.");

  File data = SD.open("datalogger.txt", FILE_WRITE);

  if (data)
  {
    data.println();

    data.println("rH (%) \t temp. (*C)");

    data.close();
  }
}

void loop()
{

```



```

    long ct = millis();

    if (ct > lst + interval)

    {

        int k=0;

        int chk_1 = DHT11.read(DHT11PIN_1);

        Serial.print("Read sensor1: ");

        switch (chk_1)

        {

        case 0: Serial.println(" SENSOR 1 IS OK");

                h[0]=DHT11.humidity;

                t[0]=DHT11.temperature;

                k++;

                break;

                break;

        case -1: Serial.println("Checksum error SENSOR 1 IS FAULTY "); k++; break;

        case -2: Serial.println("Time out error SENSOR 1 IS FAULTY "); k++; break;

        default: Serial.println("Unknown error SENSOR 1 IS FAULTY "); k++; break;

        }

        int chk_2 = DHT11.read(DHT11PIN_2);

        h[1] = 0;

        t[1] = 0;

        Serial.print("Read sensor2: ");

        switch (chk_2)

```

```

{

case 0: Serial.println(" SENSOR 2 IS OK");

        h[1]=DHT11.humidity;

        t[1]=DHT11.temperature;

        k++;

        break;

case -1: Serial.println("Checksum error SENSOR 2 IS FAULTY "); break;

case -2: Serial.println("Time out error SENSOR 2 IS FAULTY "); break;

default: Serial.println("Unknown error SENSOR 2 IS FAULTY "); break;

}

int chk_3 = DHT11.read(DHT11PIN_3);

h[2] = 0;

t[2] = 0;

Serial.print("Read sensor3: ");

switch (chk_3)

{

case 0: Serial.println(" SENSOR 3 IS OK");

        h[2]=DHT11.humidity;

        t[2]=DHT11.temperature;

        k++;

        break;

case -1: Serial.println("Checksum error SENSOR 3 IS FAULTY"); break;

case -2: Serial.println("Time out error SENSOR 3 IS FAULTY"); break;

```

```

        default: Serial.println("Unknown error SENSOR 3 IS FAULTY"); break;
    }

    int chk_4 = DHT11.read(DHT11PIN_4);

    h[3] = 0;

    t[3] = 0;

    Serial.print("Read sensor4: ");

    switch (chk_4)
    {

        case 0: Serial.println(" SENSOR 4 IS OK");

                h[3]=DHT11.humidity;

                t[3]=DHT11.temperature;

                k++;

                break;

        case -1: Serial.println("Checksum error SENSOR 4 IS FAULTY"); break;

        case -2: Serial.println("Time out error SENSOR 4 IS FAULTY"); break;

        default: Serial.println("Unknown error SENSOR 4 IS FAULTY"); break;

    }

    float hum , temp;

    hum =(h[0] + h[1] + h[2] + h[3])/ k;

    temp= (t[0] + t[1] + t[2] + t[3])/ k;

    if (maxh < hum)

    { maxh=hum;}

    if(minh > hum )

```

```

{   minh = hum; }

if(maxt < temp )

{   maxt= temp; }

if (mint > temp )

{   mint= temp;}

File data = SD.open("datalogger.txt", FILE_WRITE);

if (data) {

    data.print(hum);

    data.print("\t");

    data.println(temp);

    data.print("\t\t");


    data.print(hour());

    data.print(":");

    if(minute() < 10)

    {   data.print('0');}

    data.print(minute());

    data.print(":");

    if(second() < 10)

    {   data.print('0');}

    data.print(second());

    data.print(" ");

    data.print(day());

```

```
data.print("/");  
  
data.print(month());  
  
data.print("/");  
  
data.print(year());  
  
data.println();  
  
data.println();  
  
data.close();  
  
Serial.print(hum);  
  
    Serial.print("\t");  
  
    Serial.println(temp);  
  
    Serial.print("\t");  
  
    Serial.print(hour());  
  
    printdigits(minute());  
  
    printdigits(second());  
  
    Serial.print("\t ");  
  
    Serial.print(day());  
  
    Serial.print("/");  
  
    Serial.print(month());  
  
    Serial.print("/");  
  
    Serial.print(year());  
  
    Serial.println();  
  
    lcd.clear();  
  
    lcd.setCursor(0,0);
```

```

    lcd.print("HUMIDITY=: ");

    lcd.print(hum,DEC);

    lcd.print("% ");


    lcd.setCursor(0,1);

    lcd.print("TEMPERATURE=: ");

    lcd.print(temp,DEC);

    lcd.print("oC ");

    lst = millis();

    }

    else {

        Serial.println("error opening datalogger.txt");

        }

    }

}

void printdigits(int digits)

{

    Serial.print(":");

    if(digits < 10)

        Serial.print('0');

    Serial.print(digits);

}

```



# **CHAPTER 4**

## **RESULT AND CONCLUSION**



## **4.1 Conclusion**

Electronic notice board displaying temperature and humidity has a simple circuit where the parameters are measured using four DHT11 sensors .

Initially, circuit was selected and components where purchased and the circuit was verified in bread board. Then we designed it in vero board and the circuit was soldered onto it.

The averaged humidity and temperature values were getting displayed on the LCD screen .

The data were logged into a SD card.

If one of the sensors gets damaged then the values from the three other sensors ia averaged and displayed.

## **4.2 Future Scope and enhancements**

A system designed to measure temperature and humidity can be put in practical use, it can be used inside offices, buildings. It can also be used in an air balloon to measure the humidity and temperature of environment. It is a very simple circuit based on microcontroller, it uses a DHT11 sensor for measuring humidity and temperature. It can also be used for recording humidity and temperature values in an unattended weather station due to presence of data logging system.

In future this circuit can be improved by implementing a buzzer circuit, which will give indications when a sensor gets damaged or it shows any error. Here a SD card shield is used for data logging purposes. In place of it an ethernet shield or a wifi shield can also be used and the data logging can be done in server, which is more favourable.

# **CHAPTER 5**

## **REFERENCES**

## 5.1 References

- 1) For Arduino software <http://arduino.cc/en/main/software>
- 2) For DHT11 Libraries <http://playground.arduino.cc/main/DHT11Lib>
- 3) For DHT11 datasheet [www.micro4you.com/files/sensor/DHT11.pdf](http://www.micro4you.com/files/sensor/DHT11.pdf)
- 4) For LCD datasheet <http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>
- 5) For learning Arduino uno <http://arduino.cc/en/Main/arduinoBoardUno>
- 6) For LM35 datasheet [www.ti.com/lit/ds/symlink/lm35.pdf](http://www.ti.com/lit/ds/symlink/lm35.pdf)